

L Number	Hits	Search Text	DB	Time stamp
1	1	((cost or costs or costing) and (task or tasks) and (time or duration)) and (sum or add or summing or adding or accumulate or accumulating)) and (labor near (cost or costs))	EPO; JPO; DERWENT	2001/10/25 12:09
2	1	WO-200072211-\$.DID.	DERWENT	2001/10/25 12:10
3	15	((cost or costs or costing) and (task or tasks) and (time or duration)) and (sum or add or summing or adding or accumulate or accumulating)	EPO; JPO; DERWENT	2001/10/25 12:13
4	1	JP-2000276381-\$.DID.	DERWENT	2001/10/25 12:13

Considered all
Considered all
DP

L Number	Hits	Search Text	DB	Time stamp
2	43740	(cost or costs or costing) and (task or tasks) and (time or duration)	USPAT	2001/10/25 07:42
3	23551	((cost or costs or costing) and (task or tasks) and (time or duration)) and (sum or add or summing or adding or accumulate or accumulating)	USPAT	2001/10/25 07:51
4	647	((cost or costs or costing) and (task or tasks) and (time or duration)) and (sum or add or summing or adding or accumulate or accumulating) and (labor near (cost or costs))	USPAT	2001/10/25 07:52
5	6	((cost or costs or costing) and (task or tasks) and (time or duration)) and (sum or add or summing or adding or accumulate or accumulating) and (labor near (cost or costs)) and (non near labor)	USPAT	2001/10/25 08:05
9	690	("705/7-9").CCLS.	USPAT	2001/10/25 08:10
10	194	((("705/7-9").CCLS.) and (((cost or costs or costing) and (task or tasks) and (time or duration)) and (sum or add or summing or adding or accumulate or accumulating)))	USPAT	2001/10/25 08:11
11	19	((("705/7-9").CCLS.) and (((cost or costs or costing) and (task or tasks) and (time or duration)) and (sum or add or summing or adding or accumulate or accumulating)) and (labor near (cost or costs)))	USPAT	2001/10/25 08:11

considered all

considered all

DERWENT-ACC-NO: 2000-668523
DERWENT-WEEK: 200065
COPYRIGHT 1999 DERWENT INFORMATION LTD

TITLE: Task execution time estimation method involves performing rate simulation of area hit during interruption and accordingly calculating number of execution clocks for mean cache hit rate and execution time estimation

PATENT-ASSIGNEE: TOSHIBA KK[TOKE]

PRIORITY-DATA: 1999JP-0077679 (March 23, 1999)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE
PAGES	MAIN-IPC	
JP 2000276381	October 6, 2000	N/A
G06F 011/34		023

A

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO
APPL-DATE		
JP2000276381A	N/A	1999JP-0077679
March 23, 1999		

INT-CL_(IPC): G06F009/38; G06F011/34

ABSTRACTED-PUB-NO: JP2000276381A

BASIC-ABSTRACT: NOVELTY - Area hit rate simulation is done using interruption probability data execution trace information obtained by input program instruction level simulation, cache size, execution time, cache hit rate during interruption. Number of execution clocks is computed by adding stole penalty with trace information obtained by rate simulation, using which program execution time and mean cache hit rate are estimated.

DETAILED DESCRIPTION - The execution trace information (A) obtained as a result of instruction level simulation of input program is used in rate simulation of area hit during non-interruption and in cache simulation during interruption.

The execution trace information obtained as a result of cache simulation, is added with stole penalty for computing number of execution clocks based on which first execution time of entire program and first cache hit rate are estimated for use in rate simulation in area hit.

USE - For estimating task execution time.

ADVANTAGE - Gain of task execution time by interruption is estimated reliably, hence selection of CPU and cache system is optimized, leading to reduction in cost.

DESCRIPTION OF DRAWING(S) - The figure shows the processes involved in execution time in estimation method.

CHOSEN-DRAWING: Dwg.1/54

TITLE-TERMS:

TASK EXECUTE TIME ESTIMATE METHOD PERFORMANCE RATE SIMULATE AREA
HIT INTERRUPT

ACCORD CALCULATE NUMBER EXECUTE CLOCK MEAN CACHE HIT RATE EXECUTE
TIME ESTIMATE

DERWENT-CLASS: T01

EPI-CODES: T01-F02C; T01-F03B;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N2000-495550

【特許請求の範囲】

【請求項1】タスク実行時間を見積もる方法において、プログラム、データ初期値、実行開始番地及び実行終了番地を命令レベルシミュレーション部に入力し、分岐、命令の組み合わせを含む要因により起こるパイプラインストールを含む命令の第1の実行トレースを出力するステップと、

前記第1の実行トレースをキャッシュシミュレーション部に入力し、キャッシュミスに伴うパイプラインストールを、キャッシュ方式、キャッシュサイズに対応してシミュレーションして第2の実行トレースを出力するステップと、

前記第2の実行トレースを実行クロック数カウント部に入力し、分岐、命令の組み合わせにより起こるパイプラインストールとキャッシュミスに伴うパイプラインストールとに応じて命令あたりの実行クロック数を求めて第3の実行トレースとして出力するステップと、

キャッシュメモリとパイプライン制御部とを有する中央処理装置（以下、CPUという）の命令実行をシミュレーションするシミュレータに前記第3の実行トレースを入力し、割り込みを考慮しない場合におけるプログラム全体の第1のキャッシュヒット率及び第1の実行時間を見積もり出力するステップと、

出力された前記第1の実行トレース、前記第1のキャッシュヒット率、前記第1の実行時間、タスクの実行途中で割り込みが発生する割り込み発生率、キャッシュメモリの書き換え率を区間ヒット率シミュレーション部に入力し、前記第1の実行トレースに含まれる全ての命令と全ての割り込みとについて、キャッシュメモリ書き換えパターンに従って書き換えられたキャッシュのシミュレーションを、全ての割り込みに対して所定命令数だけ行い平均キャッシュヒット率を求めて出力するステップと、

前記第1のキャッシュヒット率と、各々の割り込みにおける前記平均キャッシュヒット率との加重平均をとって割り込みを考慮したキャッシュヒット率とする第4の実行トレースを出力するステップと、

実行クロック数カウント部に前記第4の実行トレースを入力し、各々の命令ごとにキャッシュヒット率と分岐、命令の組み合わせにより起こるパイプラインストールに応じて個々の命令ごとの実行クロック数をカウントし、第5の実行トレースとして出力するステップと、

前記第5の実行トレースを入力し、プログラム全体のキャッシュヒット率及び実行時間を見積もるステップと、割り込みを考慮した命令実行シミュレータを組み合わせることにより、割り込みを考慮した平均キャッシュヒット率及び実行時間見積もりを行うステップと、を備えたことを特徴とするタスク実行時間見積もり方法。

【請求項2】タスク実行時間を見積もる方法において、プログラム、データ初期値、実行開始番地及び実行終了

番地を命令レベルシミュレーション部に入力し、分岐、命令の組み合わせを含む要因により起こるパイプラインストールを含む命令の第1の実行トレースを出力するステップと、

前記第1の実行トレースをキャッシュシミュレーション部に入力し、キャッシュミスに伴うパイプラインストールを、キャッシュ方式、キャッシュサイズに対応してシミュレーションして第2の実行トレースを出力するステップと、

10 前記第2の実行トレースを実行クロック数カウント部に入力し、分岐、命令の組み合わせにより起こるパイプラインストールとキャッシュミスに伴うパイプラインストールとに応じて命令あたりの実行クロック数を求めて第3の実行トレースとして出力するステップと、

キャッシュメモリとパイプライン制御部とを有するCPUの命令実行をシミュレーションするシミュレータに前記第3の実行トレースを入力し、割り込みを考慮しない場合におけるプログラム全体の第1のキャッシュヒット率及び第1の実行時間を見積もり出力するステップと、
20 出力された前記第1の実行トレース、前記第1のキャッシュヒット率、前記第1の実行時間、タスクの実行途中で割り込みが発生する割り込み発生率、キャッシュメモリの書き換え率を区間ヒット率シミュレーション部に入力し、前記第1の実行トレースに含まれる命令のうち、前記割り込み発生率に比例した数の命令をサンプリングにより抽出し、この命令に対してのみ、キャッシュメモリ書き換えパターンに従って書き換えられたキャッシュのシミュレーションを行って平均キャッシュヒット率を求めて出力するステップと、

30 前記第1のキャッシュヒット率と、各々の割り込みにおける前記平均キャッシュヒット率との加重平均をとって割り込みを考慮したキャッシュヒット率とする第4の実行トレースを出力するステップと、

実行クロック数カウント部に前記第4の実行トレースを入力し、各々の命令ごとにキャッシュヒット率と分岐、命令の組み合わせにより起こるパイプラインストールに応じて個々の命令ごとの実行クロック数をカウントし、第5の実行トレースとして出力するステップと、

40 前記第5の実行トレースを入力し、プログラム全体のキャッシュヒット率及び実行時間を見積もるステップと、割り込みを考慮した命令実行シミュレータを組み合わせることにより、割り込みを考慮した平均キャッシュヒット率及び実行時間見積もりを行うステップと、を備えたことを特徴とするタスク実行時間見積もり方法。

【請求項3】請求項1又は2記載のタスク実行時間見積もり方法を、少なくともキャッシュ方式又はキャッシュサイズのいずれかが異なる複数のCPUに対してそれぞれ行う方法であって、

50 前記キャッシュシミュレーション部を用いて前記第2の

実行トレースを出力するステップにおいて、各々のCPUのキャッシュ方式及びキャッシュサイズに対して処理を行い、CPU毎の前記第2の実行トレースを求め、前記区間ヒット率シミュレーション部を用いて前記平均キャッシュヒット率を求めて出力するステップにおいて、各々のCPUのキャッシュ方式、キャッシュサイズ及び割り込み発生確率に対して処理を行い、CPU毎の前記平均キャッシュヒット率を求めることにより、CPU毎の割り込みを考慮した平均キャッシュヒット率及び実行時間を見積もり、用途に応じた最適なCPUを選択することを特徴とする実行時間見積もり方法。

【請求項4】請求項1又は2記載のタスク実行時間見積もり方法を、パイプライン段数が異なる複数のCPUに対してそれぞれ行う方法であって、

前記第3の実行トレースを出力するステップと、前記第5の実行トレースを出力するステップとにおいて、各々のCPUのストールペナルティを用いて前記第3の実行トレースと前記第5の実行トレースとを求めることにより、CPU毎の割り込みを考慮した平均キャッシュヒット率及び実行時間を見積もり、用途に応じた最適なCPUを選択することを特徴とする実行時間見積もり方法。

【請求項5】請求項1又は2記載のタスク実行時間見積もり方法を、命令セットが異なるCPUに対してそれぞれ行う方法であって、

前記第1の実行トレースを出力するステップにおいて、各々のCPUのプログラムを用いて前記第1の実行トレースを求めることにより、CPU毎の割り込みを考慮した平均キャッシュヒット率及び実行時間を見積もり、用途に応じた最適なCPUを選択することを特徴とする実行時間見積もり方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、タスク実行時間を見積もる方法に関する。

【0002】

【従来の技術】CPU (Central Processing Unit) がタスクを実行する時間を見積もる方法として、CPUの動作をシミュレーションして全命令の実行時間の合計を行う方法が一般に知られている。この方法は、パイプライン制御を行い、キャッシュメモリを有するようなCPUに対しても、パイプラインストールやキャッシュミスによる実行時間の増加を見積もることのできるという利点がある。しかし、割り込み処理に起因するキャッシュミスの増加等によって生じる実行時間の増加は、このような命令列のシミュレーションのみでは推定することができず、実行時間の見積もりの誤差要因となっていた。

【0003】

【発明が解決しようとする課題】このように、従来の実行時間の見積もり方法では、割り込みが生じた場合にはタスク実行時間の見積もり精度が低下するという問題が

あった。

【0004】本発明は上記事情に鑑みてなされたもので、割り込み処理を考慮して高い精度でタスク実行時間を見積もることが可能なタスク実行時間の見積もり方法を提供することを目的とする。

【0005】

【課題を解決するための手段】本発明のタスク実行時間の見積もり方法は、プログラム、データ初期値、実行開始番地及び実行終了番地を命令レベルシミュレーション部に入力し、分岐、命令の組み合わせを含む要因により起こるパイプラインストールを含む命令の第1の実行トレースを出力するステップと、前記第1の実行トレースをキャッシュシミュレーション部に入力し、キャッシュミスに伴うパイプラインストールを、キャッシュ方式、キャッシュサイズに対応してシミュレーションして第2の実行トレースを出力するステップと、前記第2の実行トレースを実行クロック数カウント部に入力し、分岐、命令の組み合わせにより起こるパイプラインストールとキャッシュミスに伴うパイプラインストールとに応じて命令あたりの実行クロック数を求めて第3の実行トレースとして出力するステップと、キャッシュメモリとパイプライン制御部とを有するCPUの命令実行をシミュレーションするシミュレータに前記第3の実行トレースを入力し、割り込みを考慮しない場合におけるプログラム全体の第1のキャッシュヒット率及び第1の実行時間を見積もり出力するステップと、出力された前記第1の実行トレース、前記第1のキャッシュヒット率、前記第1の実行時間、タスクの実行途中で割り込みが発生する割り込み発生率、キャッシュメモリの書き換え率を区間ヒット率シミュレーション部に入力し、前記第1の実行トレースに含まれる全ての命令と全ての割り込みとについて、キャッシュメモリ書き換えパターンに従って書き換えられたキャッシュのシミュレーションを、全ての割り込みに対して所定命令数だけ行い平均キャッシュヒット率を求めて出力するステップと、前記第1のキャッシュヒット率と、各々の割り込みにおける前記平均キャッシュヒット率との加重平均をとって割り込みを考慮したキャッシュヒット率とする第4の実行トレースを出力するステップと、実行クロック数カウント部に前記第4の実行トレースを入力し、各々の命令ごとにキャッシュヒット率と分岐、命令の組み合わせにより起こるパイプラインストールに応じて個々の命令ごとの実行クロック数をカウントし、第5の実行トレースとして出力するステップと、前記第5の実行トレースを入力し、プログラム全体のキャッシュヒット率及び実行時間を見積もるステップと、割り込みを考慮した命令実行シミュレータを組み合わせることにより、割り込みを考慮した平均キャッシュヒット率及び実行時間見積もりを行うステップとを備えたことを特徴としている。

【0006】また、本発明のタスク実行時間を見積もる

方法は、特定の命令をサンプリングにより抽出して実行時間を見積もる方法であって、プログラム、データ初期値、実行開始番地及び実行終了番地を命令レベルシミュレーション部に入力し、分岐、命令の組み合わせを含む要因により起こるパイプラインストールを含む命令の第1の実行トレースを出力するステップと、記第1の実行トレースをキャッシュシミュレーション部に入力し、キャッシュミスに伴うパイプラインストールを、キャッシュ方式、キャッシュサイズに対応してシミュレーションして第2の実行トレースを出力するステップと、前記第2の実行トレースを実行クロック数カウント部に入力し、分岐、命令の組み合わせにより起こるパイプラインストールとキャッシュミスに伴うパイプラインストールとに応じて命令あたりの実行クロック数を求めて第3の実行トレースとして出力するステップと、キャッシュメモリとパイプライン制御部とを有するCPUの命令実行をシミュレーションするシミュレータに前記第3の実行トレースを入力し、割り込みを考慮しない場合におけるプログラム全体の第1のキャッシュヒット率及び第1の実行時間を見積もり出力するステップと、出力された前記第1の実行トレース、前記第1のキャッシュヒット率、前記第1の実行時間、タスクの実行途中で割り込みが発生する割り込み発生率、キャッシュメモリの書き換え率を区間ヒット率シミュレーション部に入力し、前記第1の実行トレースに含まれる命令のうち、前記割り込み発生率に比例した数の命令をサンプリングにより抽出し、この命令に対してのみ、キャッシュメモリ書き換えパターンに従って書き換えられたキャッシュのシミュレーションを行って平均キャッシュヒット率を求めて出力するステップと、前記第1のキャッシュヒット率と、各々の割り込みにおける前記平均キャッシュヒット率との加重平均をとって割り込みを考慮したキャッシュヒット率とする第4の実行トレースを出力するステップと、実行クロック数カウント部に前記第4の実行トレースを入力し、各々の命令ごとにキャッシュヒット率と分岐、命令の組み合わせにより起こるパイプラインストールに応じて個々の命令ごとの実行クロック数をカウントし、第5の実行トレースとして出力するステップと、前記第5の実行トレースを入力し、プログラム全体のキャッシュヒット率及び実行時間を見積もるステップと、割り込みを考慮した命令実行シミュレータを組み合わせることにより、割り込みを考慮した平均キャッシュヒット率及び実行時間見積もりを行うステップとを備えたことを特徴とする。

【0007】また、タスク実行時間の見積もりを、少なくともキャッシュ方式又はキャッシュサイズのいずれかが異なる複数のCPUに対してそれぞれ行う場合は、前記キャッシュシミュレーション部を用いて前記第2の実行トレースを出力するステップにおいて、各々のCPUのキャッシュ方式及びキャッシュサイズに対して処理を

行い、CPU毎の前記第2の実行トレースを求め、前記区間ヒット率シミュレーション部を用いて前記平均キャッシュヒット率を求めて出力するステップにおいて、各々のCPUのキャッシュ方式、キャッシュサイズ及び割り込み発生確率に対して処理を行い、CPU毎の前記平均キャッシュヒット率を求めることにより、CPU毎の割り込みを考慮した平均キャッシュヒット率及び実行時間を見積もり、用途に応じた最適なCPUを選択する。

【0008】あるいは、タスク実行時間見積もりを、パイプライン段数が異なる複数のCPUに対してそれぞれ行う場合は、前記第3の実行トレースを出力するステップと、前記第5の実行トレースを出力するステップとにおいて、各々のCPUのストールペナルティを用いて前記第3の実行トレースと前記第5の実行トレースとを求めることにより、CPU毎の割り込みを考慮した平均キャッシュヒット率及び実行時間を見積もり、用途に応じた最適なCPUを選択する。

【0009】さらには、タスク実行時間見積もりを、命令セットが異なるCPUに対してそれぞれ行う方法場合には、前記第1の実行トレースを出力するステップにおいて、各々のCPUのプログラムを用いて前記第1の実行トレースを求めることにより、CPU毎の割り込みを考慮した平均キャッシュヒット率及び実行時間を見積もり、用途に応じた最適なCPUを選択する。

【0010】

【発明の実施の形態】以下、本発明の実施の形態について図面を参照して説明する。

【0011】本発明の第1の実施の形態によるタスク実行時間の見積もり方法は、図1に示されるような構成を備えており、割り込みを考慮しない見積もり方法と割り込みを考慮した見積もり方法とを組み合わせている。そこで、第1の実施の形態の説明を行う前に、図2に示された割り込みを考慮しない見積もり方法と、図25に示された割り込みを考慮した見積もり方法とをそれぞれ分けて説明する。いるが、先ず、割り込みを考慮したキャッシュヒット率及びタスク実行時間を求めることができる原理について述べる。

【0012】図3に、割り込みを考慮しない場合におけるタスク実行中のキャッシュヒット率を示す。ここで、キャッシュヒット率12001は少数の命令を実行するときの平均キャッシュヒット率であり、キャッシュヒット率12002は十分に多数の命令を実行するときの平均キャッシュヒット率である。この図3より、キャッシュヒット率は、タスクの開始直後から上昇してある一定値で安定することが分かる。ここで、キャッシュヒット率の上昇率は、用いられるアプリケーションプログラムの内容に依存し、安定したときの値はキャッシュサイズとプログラムの内容に依存する。

【0013】図4に、タスクの実行中に割り込みが発生し、割り込みから復帰した場合におけるキャッシュヒッ

ト率の変化を示す。この図4において、キャッシュヒット率13001及び13002は、それぞれ図3におけるキャッシュヒット率12001及び12002に対応する。ここで、タスクの実行途中で割り込みが起こると、割り込みルーティンによりキャッシュメモリの書き換えが起こる。このため、割り込みルーティンからの復帰直後では、タスクの実行開始時と同様にキャッシュヒット率が低下する。しかし、復帰後から十分な数の命令が実行されると、キャッシュヒット率13003及び13004は安定する。

【0014】割り込みルーティンから復帰した直後はキャッシュヒット率が低下するため、命令の実行時間が増加し、タスクの総実行時間も増加する。この実行時間の増加は、タスクの性質やキャッシュサイズだけでなく、割り込みの発生率や割り込みルーティンによるキャッシュメモリの書き換え状況によっても異なる。このため、ただ単にタスクの実行時間を見積もるだけでは予測することができない。

【0015】このことを、図5～図7を用いてさらに説明する。割り込みルーチンが長く、殆どのキャッシュデータが書き替えられた場合のキャッシュヒット率は、図5に示された通りである。割り込みルーチン処理から復帰した直後のキャッシュヒット率14004及び14005は、ほぼ0まで低下する。

【0016】割り込みルーチンが短く、多くのキャッシュデータが破壊されずに残存し、一部のデータが新しく置き変わった場合のキャッシュヒット率は、図6に示された通りである。割り込みルーチン処理から復帰した直後のキャッシュヒット率15004及び15005は、図5の場合と異なり低下はするが0とはならない。また、割り込み発生率によって実効キャッシュヒット率は異なり、発生率が高ければ実効ヒット率16003は大きく低下し、発生率が低ければ実効ヒット率16002はあまり低下しない。

【0017】このことから、タスクの実行時間は、割り込み発生率と割り込みルーティンのキャッシュ書き換える程度によって変化することが分かる。

【0018】以上の点を考慮し、割り込みによるキャッシュヒット率の低下を見積もる手順を、図8を用いて説明する。図8に、ある命令の実行時点Sにおけるキャッシュヒット率17001と、割り込みを考慮しない場合のタスク全体のキャッシュの平均キャッシュヒット率A 17002とを示す。実行時点Sにおいて割り込みが発生し、割り込みルーティンが実行された後、もとのタスクに復帰するとキャッシュヒット率17003が上昇し、平均ヒット率A 17002まで戻ると仮定する。このように考えると、実行時点Sにおけるキャッシュヒット率17001からある一定命令数の平均キャッシュヒット率Bi 17004を求めることができる。

【0019】この平均キャッシュヒット率Bi 170

04を、全ての割り込みに対して求め、割り込みが発生しない場合の平均キャッシュヒット率A 17002との間で加重平均をとることで、実行時点Sにおける割り込みを考慮した平均キャッシュヒット率Hを求めることができる。

【0020】この平均キャッシュヒット率Hをすべての命令に対して求め、それに基づいて実行クロック数を求めることにより、割り込みを考慮した実行時間を求めることが可能となる。

10 【0021】以下に、割り込みを考慮しないタスク実行時間の見積もり方法の処理の手順と、割り込みを考慮した実行時間の見積もりの処理方法の手順とを比較しながら説明する。

【0022】図2に、割り込みを考慮しない場合のタスク実行時間の見積もり方法のフローチャートを示す。

【0023】プログラム、データ初期値、実行開始番地及び終了番地1001を命令レベルシミュレーション1002に入力し、プログラムの実行トレース情報A 1003を出力する。

20 【0024】命令レベルシミュレーション1002から出力された実行トレース情報A 1003と、キャッシュ方式及びキャッシュサイズに関するデータ1004とをキャッシュシミュレーション1005に入力し、実行トレース情報B1 1006を出力する。

【0025】このトレース情報B1 1006と、ストールペナルティ情報1007とを入力として、各命令毎の実行クロック数を求める処理1008を行い、実行トレース情報C1 1009を出力する。

30 【0026】この実行トレース情報C1 1009を入力とし、タスク全体のキャッシュヒット率及び実行時間を見積もる処理1010を行い、その見積もり結果1011を出力する。

【0027】次に、割り込みを考慮した場合のタスク実行時間の見積もり方法のフローチャートを図25に示す。

【0028】実行時間を見積もるプログラム、データ初期値、実行開始番地及び終了番地に関するデータ2001を入力として、命令レベルシミュレーション処理2002を行い、プログラムの実行トレース情報A 2003を出力する。

40 【0029】実行トレース情報A 2003、キャッシュ方式及びキャッシュサイズに関するデータ2004、割り込み確率データ2005を入力として、区間ヒット率シミュレーション処理2006を行い、各命令でのキャッシュヒット率を含む実行トレース情報B2 2007を出力する。

【0030】出力された実行トレース情報B2 2007と、ストールペナルティ情報 2008とを入力して、各命令毎の実行クロック数を求めて実行トレース情報C22010を出力する。この実行トレース情報C2

2010を用いてキャッシュヒット率と実行時間とを見積もる処理2011を行い、見積もり結果2012を出力する。

【0031】図2に示された割り込みを考慮しない見積もり処理と、図25に示された割り込みを考慮した見積もり処理とを対比すると、図2に示された命令レベルシミュレーション処理1002と図25に示された命令レベルシミュレーション処理2002とは処理内容が同一で、入力及び出力データとも同一である。しかし、処理1002、2002からそれぞれ出力されたトレース情報A 1003、2003を用いた以降の処理は相違している。

【0032】まず、命令レベルシミュレーション処理1002、2002の詳細な処理の内容は、図9～図13のフローチャートに示されるようである。ここで、命令レベルシミュレーション処理の説明を行う前に、この処理によって出力される実行トレース情報の一例を図14～図16に示す。図14に示されるように、出力される実行トレース情報には命令の先頭番地3001、実行した命令3002、分岐やデータアクセスのための実効番地3003、分岐が行われたことを示す分岐フラグ3004が含まれる。対象とするCPUによっては実効番地3003が複数存在する場合があり、そのような場合には全ての番地を含めるものとする。

【0033】対象とするCPUが、分岐以外にもパイプラインストールを起こす場合には、図14に示された情報3001～3004から成る図14の情報4001の他に、ストールを起こしたことを示すストールフラグ4002を追加する必要がある。

【0034】さらに、対象とするCPUにおいて、命令やデータの語長が可変である場合には、図14に示された情報3001～3004と、これに図15に示される情報4002とを加えた図15の情報5001の他に、命令アクセス回数5002、データアクセス回数5003に関する情報を追加する必要がある。

【0035】このような出力を行う命令レベルシミュレーション処理の内容について図9を参照して述べる。図2又は図25に示された入力データ1001又は2001に従ってメモリを初期化し、プログラムの各命令に対して処理21001～21008を行う。尚、ここではCPUはパイプライン処理により命令を実行し、分岐以外ではストールは発生しないと仮定する。

【0036】まず、処理21001として分岐フラグをリセットし、処理21002として1つの命令を読み込む。次に、命令の種別、即ち分岐命令、ロード命令、セーブ命令、レジスタ間演算命令か否かを処理21003～21006においてそれぞれ判断し、分岐命令であれば処理21009としてこの命令を実行し、ロード命令であれば処理21010としてこの命令を実行し、セーブ命令の場合にはこの命令を処理21011として実行

し、レジスタ間演算命令であればこの命令を処理21012として実行する。上記命令のいずれにも該当しない命令に対しては、処理21013としてその命令を実行する。そして、各々の命令を実行した結果に基づいて、処理21007としてトレース情報を出力する。以上の処理21001～21007を、処理21008として示されたように、各命令の実行により更新されたプログラムカウンタPCが終了番地と等しくなるまで繰り返す。次に、命令毎の処理21009～21013について、図10～13を用いて詳述する。分岐命令の処理21009は、図10に示されたように、先ず分岐が起こるかどうかを処理22001としてを判断する。分岐が起こらない場合は、処理22003としてプログラムカウンタPCに現在の命令の番地の次の番地を設定する。分岐が起こる場合は、処理22002として分岐先の実効番地を計算する。処理22004としてプログラムカウンタPCに実行番地を書き込み、処理22005として分岐フラグをセットする。

【0037】ロード命令の実行処理21010は、メモリからのロード命令を実行する処理であり、処理の内容は図11に示されるようである。処理23001としてメモリの実効番地を計算し、この命令が示すレジスタから、計算した実効番地の値を処理23002として読み込み、処理23003としてプログラムカウンタを次の命令の番地にセットする。

【0038】セーブ命令の実行処理21011は、メモリへのセーブ命令を実行するための処理であってその処理の内容は図12に示されるようである。先ず、メモリの実効番地を処理24001として計算し、処理24002として命令が示すレジスタにその番地の値を書き込む。そして、処理24003としてプログラムカウンタPCを次の命令の番地にセットする。

【0039】レジスタ間演算命令の実行処理21012の処理内容は、図13に示されるようである。処理25001としてソースレジスタ1の値を読み、処理25002としてソースレジスタ2の値を読む。読み込んだ二つの値の間で、命令が示す演算を処理25003として実行し、その結果を処理25004としてディストネーションレジスタに書き込む。処理25005として、プログラムカウンタPCを次の命令の番地にセットする。

【0040】他の命令の実行処理21003に対して、上記処理21009～21011と同様に、実際のCPUが実行するように処理を行うことで、CPUが実行すべき全ての命令に対してシミュレーションを行うことができる。

【0041】図2におけるキャッシュシミュレーション処理1005と、図25に示された区間ヒット率シミュレーション処理2006とは、同一の実行トレース情報A1003又は2003に対する処理である。しかし、キャッシュシミュレーション処理1005では、割り込

みの影響を考えないキャッシュのシミュレーションを行い、区間ヒット率シミュレーション処理2006では割り込みの影響を考えたキャッシュのシミュレーションを行う点が異なる。

【0042】割り込みの影響を考えないキャッシュシミュレーション処理1005は、図20に示されるようなダイレクトマップ、ライトバック方式のキャッシュメモリに対して行うシミュレーション処理である。キャッシュメモリの方式としてはこの他に、セットアソシアティブ、フルアソシアティブ等が知られているが、これらの方式であってもその方式に従ったシミュレーションを行うことができる。

【0043】まず、命令とデータに対してそれぞれ別のキャッシュメモリを使用すると仮定する。図20に示されたキャッシュメモリは、アドレスが32ビット、データが32ビット、キャッシュメモリが16Kエントリという構成を備えている。このキャッシュメモリは、1ビットの有効データ30008と、16ビットのタグデータ30009と、32ビットのデータ30010とを備え、16Kワードのメモリ30005と16ビットの比較器30006とANDゲート30007とにより構成される。

【0044】データ30010における32ビットのアドレスは、上位からデータ30001分の16ビット、データ30002分の14ビット、データ30004分の2ビットに分割されている。このうち、データ30002をインデックスとしてメモリをアクセスして得られるタグデータ30009の値がデータ30001の16ビットと等しく、かつ有効データ30008の値が「1」であるときにデータ30010が指定されたアクセスのデータとなる。

【0045】このような構成を有するメモリを対象としてキャッシュシミュレーションを行うために、図21で示されるような有効データ31003とタグM31002から構成されるデータ構造31001を使用する。この場合のキャッシュシミュレーションの処理の手順について、全体の流れを示した図22と、リードアクセスの流れを示した図23と、ライトアクセスの流れを示した図24とを用いて説明する。

【0046】分岐命令、ロード命令、セーブ命令の各命令毎に、実効番地がキャッシュメモリに保持されているか否かを判断して、その結果をトレース情報として出力する。まず、処理26001として実行トレースから1つの命令に関する情報を読み込む。処理26002として命令があるか否かを判断し、ない場合にはこの時点で終了する。命令が存在する場合には、処理26003として分岐命令キャッシュのリードアクセスを行う。処理26009としてヒットしたか否かを判断し、ヒットした場合には処理26010として命令キャッシュヒットフラグをセット(On)し、ヒットしなかった場合には処

理26011としてリセット(Off)する。

【0047】処理26004として命令がロード命令か否かを判断し、処理26012においてデータキャッシュのリードアクセスを行う。処理26013としてヒットしたか否かを判断し、ヒットした場合には処理26014としてデータキャッシュヒットフラグをセットし、ヒットしなかった場合には処理26015としてリセットする。

【0048】処理26005として命令がストア命令か否かを判断し、処理26016においてデータキャッシュのライトアクセスを行う。処理26017としてヒットしたか否かを判断し、ヒットした場合には処理26018としてデータキャッシュヒットフラグをセットし、ヒットしなかった場合には処理26019としてリセットする。

【0049】以上の処理26001～26019を全て命令に対して行っていく、処理26007として処理の結果をトレース情報として出力する。

【0050】ここで、キャッシュメモリのリードアクセス処理26003について、図23を用いて詳述する。処理28001として、実効アドレスをタグ、インデックス、バイトオフセットに分割し、インデックスで指定される有効値が「1」か否かを処理28002で判断する。この値が「1」ではない場合に、さらに処理28003においてインデックスで指定されるタグMの値がタグに等しいか否かを判断する。この値が等しい場合に、キャッシュがヒットしたことになる。インデックスで指定される有効値が「1」ではない場合、及び／又はインデックスで指定されるタグMの値がタグに等しくない場合はヒットせずキャッシュミスであった場合に相当し、処理28004として有効値とタグMの値とを更新する。実際のシステムにおいては、ヒットしなかった場合には外部のメモリからのデータの読み込みが必要となるが、ここではこの動作をシミュレーションする必要はない。キャッシュメモリのライトアクセス処理26016について、図24を用いて詳述する。まず、実効番地をタグ、インデックス、バイトオフセットに分割し(29001)、インデックスで指定される有効値が1であり(29002)、インデックスで指定されるタグMの値がタグに等しいとき(29003)、キャッシュがヒットしたことになる。そうでないときはキャッシュミスであるため、有効とタグMの値を更新する(29004)。現実のシステムではキャッシュミスとしときは外部のメモリからのデータの書き込み(ライトバック)が必要となるが、この部分のシミュレーションは必要ない。

【0051】図22～図24に示される処理を命令列に対して行うことで、実行トレースで示される番地の命令についてキャッシュシミュレーションを行うことができる。また、上記メモリと異なる方式のキャッシュメモリ

に対しても同様の方法でシミュレーションを行うことができる。

【0052】この処理の結果は、図2に示された実行トレース情報B1 1006として出力され、その内容は図17に示されるように、実行トレース情報A 6001に命令キャッシュヒット回数6002とデータキャッシュヒット回数6003とが付加されたものになっている。

【0053】次に、割り込みを考慮した場合の図25に示された区間ヒット率シミュレーション処理2006について説明する。ここで、実行時間の見積もりを行うタスクについては、図2に示された割り込みを考慮しない場合におけるキャッシュヒット率が、それぞれ命令及びデータに対して求められているものとする。また、以下に説明する見積もり方法は、命令、データキャッシュについても同様に適用することができる。

【0054】図26及び図27に、区間ヒット率シミュレーション処理1006の手順を示す。この処理は、実行トレースに現れる全ての命令を実行する時に、予め定められた確率で割り込みが起こると仮定して、平均キャッシュヒット率を求めるものである。区間ヒット率シミュレーション処理1006を、実行トレースのすべての命令に対して行う場合は、図26に示されたような手順で行う。ある特定の命令を抽出し、その命令に対してのみ区間ヒット率シミュレーションを行う場合は、後述する第2の実施の形態のように図40に示された手順に従う。

【0055】即ち、処理34001として、実行トレース中の全ての命令について処理したか否かを判断し、処理34002として存在するすべての割り込みについて区間ヒット率を求めたか否かを判断する。処理34002において区間ヒット率を求めている割り込みがまだ存在すると判断した場合は、処理34003として区間ヒット率を求める。全ての割り込みに対して区間ヒット率を求め終わった場合は、処理34004として、割り込みを考慮しない場合の平均ヒット率と割り込みを考慮した場合の区間ヒット率とで加重平均をとり、当該命令のキャッシュヒット率とする。

【0056】ここで、処理34003として区間ヒット率を求める手順は、図27に示されたようである。処理36001として、キャッシュを完全にクリアしたか否かを判断し、完全にクリアしていない場合は処理36002としてアクセスパターンが存在するか否かを判断し、アクセスパターンが存在する場合は次の処理36004へ移行する。

【0057】処理36004として、アクセスパターンにあるインデックスの有効フラグをオフにする。処理36007として、アクセスパターンにないインデックスの有効「0」フラグをオフにして処理36005へ移行する。ここで、上記処理36002においてアクセスパ

ターンが存在しない場合は、処理36005へ直ちに移行する。

【0058】処理36005として、乱数に従ってキャッシュの有効フラグと有効「0」フラグとをオフにする。また、上記処理36001において、キャッシュが完全にクリアされている場合には、処理36003としてキャッシュの全ての有効フラグをオフにし、処理36006としてキャッシュの全ての有効「0」フラグをオフにした後、処理36008へ移行する。

【0059】処理36008として、定められた命令数についてキャッシュシミュレーションを行う。処理36009として、平均ヒット率を求める。

【0060】この区間ヒット率を求める処理は、割り込み要因別に例えば図28に示されたように分類して行うことができる。割り込みの種別39001としてA～Eがあるとする、それぞれの発生確率39002は2.0×10⁻⁵、1.5×10⁻⁶、2.0×10⁻⁷、2.0×10⁻⁶、1.0×10⁻⁷であり、キャッシュの書き換わるパターン39003はそれぞれ30%、パターンA、パターンB、100%クリア、80%クリアとする。書き換わりパターンAは図29、書き換わりパターンBは図30にそれぞれ示されるようであるとする。

【0061】一般に、キャッシュの書き換わるパターンは、割り込み処理を行うルーティンの性質によって異なる。例えば、割り込み処理が長い場合には命令キャッシュ、データキャッシュともに殆どの部分を書き換わってしまうと考えられる。このような場合には、図28において割り込み種別Dとして示されたように、100%クリアされると考えてよい。このクリアされる確率は、ルーティンの長さ等によって変化し、これらは予め割り込みルーティンを解析することで求められているものとする。また、このようなクリアされる確率を求めるための解析は、上述した命令レベルシミュレーションとキャッシュシミュレーションとを用いて行うことができる。

【0062】処理ルーティンの内容によっては、図29に示された上記書き換わりパターンA、あるいは図30に示された書き換わりパターンBのように、キャッシュの特定の部分のみが書き換わる場合も考えられる。このような場合は、図29、図30にそれぞれ示されたように、書き換わるインデックスXの値を列挙することで書き換えのパターンを指定することが可能である。

【0063】また、図28に示された上記割り込み種別Aと割り込み種別Bとを組み合わせる方法によって、キャッシュの書き換わりパターンを表すこともできる。これは、例えば特定番地のデータとスタック上の作業領域とを使用する処理等において見られる書き換わりパターンであって、図28の割り込み種別Cのように示すことができる。

【0064】そして、割り込み種別A～Dとして示されたようなキャッシュの書き換わりパターンで書き換わり

の起こるキャッシュのシミュレーションは、図21に示されたようなキャッシュのデータ構造、及び図23、図24に示されたようなキャッシュのアクセス方法では行うことができない。そこで、図31に示されるようなキャッシュのデータ構造、及び図32、図33に示されたようなキャッシュのアクセス方法でキャッシュシミュレーションを行う必要がある。

【0065】図31に示されたデータ構造は、書き換わりを考慮したものである。このデータ構造は、図21に示されたデータ構造と同様に、有効54001とタグM54003とを含んでいる他に、さらに、有効「0」54002を含んでいる。この有効「0」54002は、割り込みルーティンにより書き換えが生じていないエントリについて「0」にセットされるフラグである。このフラグは、そのエントリがアクセスされた場合には、有効54001とタグM54003との書き換えは起こるが、キャッシュはヒットしたものととして扱うために設けられている。この有効「0」54002が付加されたことで、書き換わったエントリ以外のエントリにタグが設定されていなくとも、キャッシュがヒットしたものととして扱うことが可能である。

【0066】このようなデータ構造を有するキャッシュをアクセスする手順を、図32、図33に示す。リードアクセスについての処理は図32、ライトアクセスの処理は図33に示されるとおりである。この図32に示された処理55001～55006と、図23を用いて説明した割り込みを考慮しない場合のリードアクセスの処理28001～28004とを比較すると、処理55001、55003～55005と処理28001～28004とが同一の内容であることがわかる。そして、図32に示された処理55002及び55006は、有効「0」フラグに関して新たに付加された処理であって、有効「0」フラグが「0」のときにエントリが書き換わる場合にヒットとして扱うように構成されている。また、有効「0」フラグは1回アクセスがあるとその後は「1」となる。よって、2回目以上のアクセスでは図23に示された割り込みを考慮しない場合のリードアクセスと同様の処理を行う。

【0067】リードアクセスの場合と同様に、ライトアクセスの処理56001～56006のうち処理56001、56003～56005は、図24に示された割り込みを考慮しない場合のライトアクセスの処理29001～29004と同様である。そして、有効「0」フラグに関する処理56002及び56006が新たに付加されている。

【0068】そして、区間ヒット率は図27で示されるような手順で求められる。処理36003～36005により、キャッシュの書き換わりパターンに従って書き換わるエントリに関して有効フラグを「0」に設定し、処理36005～36007によって、書き換わらない

エントリについては有効「0」フラグを「0」にする。このような処理により、実質的にヒット率は低下する。

【0069】次に、処理36008として、予め実験等を行って定めておいた命令数についてキャッシュシミュレーションを行う。この方法は、図22、図32及び図33に示される手順でキャッシュシミュレーションを行い、処理36009として当該命令群について平均キャッシュヒット率を求めて区間ヒット率とする。

【0070】以上のように、区間ヒット率シミュレーションにおけるキャッシュシミュレーションは、図22、図32、図33に示されたように行う点を除いて、割り込みを考慮しない場合と同様な手順でキャッシュアクセス処理を行うことができる。得られた区間ヒット率シミュレーションは、図25に示されたように、実行トレース情報B2 2007として出力される。この情報B2 2007の内容は、割り込みを考慮しない場合と同様に、図17に示されたような実行トレースA6001に、命令キャッシュヒット回数6002とデータキャッシュヒット回数6003とが追加されたものになっている。割り込みを考慮しない場合との相違は、命令キャッシュヒット回数6002とデータキャッシュヒット回数6003との値が整数ではなく、キャッシュヒット率によって定まる実数となる点である。

【0071】次に、実行クロック数を求める処理について述べる。この処理は、図2に示された割り込みを考慮しない場合の処理1008と、図25に示された考慮しない場合の処理2009とでは、キャッシュヒットの扱いが異なる。割り込みを考慮しない場合の処理は図37に示すようであり、割り込みを考慮する場合の処理は図34に示すようである。

【0072】割り込みを考慮しない場合は、図35(a)に示されたような仕様を有するCPU Aを対象とする。このCPU Aは、図36(a)に示されたように、分岐によるパイプラインストールのペナルティNBRは5クロック、命令キャッシュミスによるペナルティNIFは3クロック、データキャッシュミスによるペナルティNDATAは3クロック、命令の組み合わせによるストールNISTLは起こらないものとする。そして、これらのペナルティは図2に示されたストールペナルティ1007に相当する。

【0073】このような仕様のCPU Aに対して、図37に示されたような処理32001～32009を行っていく。まず、処理32001として実行に必要な最低クロック数NCを1とし、処理32002～32005としてそれぞれ分岐によるストール、命令の組み合わせによるストール、命令キャッシュミスによるストール、データキャッシュミスによるストールが発生したか否かを、実行トレース情報に含まれる情報を用いて判断する。そして、それぞれの場合においてストールが発生したときは処理32006～32009としてNCの値

をストールペナルティとして与えられる値だけ加算する。

【0074】このような処理32001～32009によって得られた値を実行トレースB1 1006に追加したものが実行トレースC1 1009である。この情報は、図18に示されたようにトレース情報B1の命令毎の内容7001に、命令毎の実行クロック数7002が付加されたものとなっている。

【0075】次に、割り込みを考慮した場合の実行クロック数を求める処理を図34を用いて説明する。処理37001として、実行に必要な最低のクロック数NCの設定を行い、処理37002及び37007として分岐によるストールに関する処理を行い、処理37003及び37008として命令の組み合わせによるストールに関する処理を行う。この処理は、割り込みを考慮しない場合の図37に示された処理32001、32002及び32006、32003及び32007とそれぞれ同一である。

【0076】ところが、割り込みを考慮した場合には、キャッシュメモリのヒット率を実数で表される。このため、命令キャッシュミスとデータキャッシュミスに関しては、それぞれ処理37004及び37005においてキャッシュヒット率に基づいてペナルティとなるクロック数を計算する。この演算は、命令キャッシュミスによるペナルティNIFと、データキャッシュミスによるペナルティNDATAにキャッシュミスの確率をかけたものをクロック数NCに加算すればよい。

【0077】さらに、割り込みの発生の際には確実にバイブラインストールが発生する。このため、処理37005として、割り込み発生率にストールペナルティを掛けたものを加算する必要がある。ここで、割り込みの際のストールペナルティの値は分岐によるペナルティと同じ値であると仮定しているが、CPUの仕様によっては異なる値となる場合がある。そのような場合にはストールペナルティ2008に、この値をあらかじめ定義しておけばよい。

【0078】このような処理によって得られたストールペナルティ2008を実行トレースB2 2007に加えることにより、図25に示されたように実行トレース情報C2 2010が得られる。この情報C2 2010は、割り込みを考慮しない場合と同様に、図18に示されたように、トレース情報B2の命令毎の内容7001に、命令毎の実行クロック数7002が付加されている。割り込みを考慮しない場合における情報との相違は、実行クロック数が実数となる点である。

【0079】キャッシュヒット率と実行時間の見積もりを行う処理は、割り込みを考慮する場合の処理2011と、考慮しない場合の処理1010とで異なる。割り込みを考慮する場合の処理2011の内容を図39、考慮しない場合の処理1010を図38にそれぞれ示す。こ

の処理2011と1010との相違は、個々の命令におけるキャッシュヒットを回数でとらえるか、あるいは確率でとらえるかという点にある。

【0080】割り込みを考慮しない場合の処理1010は、図38の処理33001として示されたように、実行トレースから1命令の情報を読み込む毎に以降の処理を進めていく。処理33003として命令数を1増やし、処理33004として実行トレースに従って命令アクセス回数を加算し、処理33005としてデータアクセス回数を加算し、処理33006として実行クロック数を加算する。

【0081】次に、処理33007として命令キャッシュがヒットしたか否かを判断し、ヒットした場合には処理33009として命令キャッシュヒット回数を更新する。処理33008として、データキャッシュがヒットしたかどうかを判断し、ヒットした場合は処理33010としてデータキャッシュヒット回数を更新する。全ての命令について処理33001～33010を行い、処理33011として、命令キャッシュヒット回数が命令アクセス回数に占める比率として命令キャッシュヒット率を求める。処理33012として、データキャッシュヒット回数がデータアクティブ回数に占める比率としてデータキャッシュヒット率を求めて終了する。これらの値を、見積もり結果1011として出力する。

【0082】以下に、割り込みを考慮した場合の処理を図39を用いて説明する。この処理は、実行クロック数、命令、データキャッシュのヒット回数を実数である点を除いて、図38を用いて説明した割り込みを考慮しない場合の処理と同様である。即ち、図39に示された処理38001～38005と、図38に示された処理33001～33005とは同一であり、命令数、命令アクセス回数、データアクセス回数が同様に求められる。

【0083】しかし、図38における処理33006と異なり、図39における処理38006において扱う実行クロック数は実数であるため、この演算結果は実数で表現される。また、図38における処理33009と異なり、図39に示された処理38007における命令キャッシュヒット率の演算は、命令アクセス回数にその命令におけるキャッシュヒット率を考慮し、命令アクセス回数にキャッシュヒット率を乗じたものを加算する。さらに、図38における処理33010と異なり、図39における処理38008は、データアクセス回数にその命令におけるキャッシュヒット率を考慮して、データアクセス回数にキャッシュヒット率を乗じたものを加算する。

【0084】そして、図38に示された処理33011、33012と同様に、図39に示された処理38009では命令キャッシュヒット回数が命令アクセス回数に占める比率として命令キャッシュヒット率を求め、処

理38010ではデータキャッシュヒット回数がデータアクセス回数に占める比率としてデータキャッシュヒット率を求める。このようにして得られた値を、見積もり結果2012として出力する。

【0085】以上説明した図2に示された割り込みを考慮しない場合の見積もり方法と、図25に示された割り込みを考慮した場合の見積もり方法とを組み合わせた本発明の第1の実施の形態について、図1を用いて説明する。

【0086】プログラム、データ初期値、実行開始番地及び終了番地1001を命令レベルシミュレーション1002に入力し、プログラムの実行トレース情報A 1003を出力する。

【0087】命令レベルシミュレーション1002から出力された実行トレース情報A 1003と、キャッシュ方式及びキャッシュサイズに関するデータ1004とをキャッシュシミュレーション1005に入力し、実行トレース情報B1 1006を出力する。

【0088】このトレース情報B1 1006と、ストールペナルティ情報1007とを入力として、各命令毎の実行クロック数を求める処理1008を行い、実行トレース情報C1 1009を出力する。

【0089】この実行トレース情報C1 1009を入力とし、タスク全体のキャッシュヒット率及び実行時間を見積もる処理1010を行い、その見積もり結果1011を出力する。

【0090】さらに、得られた見積もり結果1011と、実行トレース情報A 1003、キャッシュ方式及びキャッシュサイズに関するデータ2004、割り込み確率データ2005を入力として、区間ヒット率シミュ

$$(1/50 * 106) / 2 * 10^{-3} = 1 * 10^{-5} \quad \dots (1)$$

となる。そこで、第2の実施の形態では個々の命令で全ての種類の割り込みについての平均ヒット率を求めるのではなく、全実行区間から割り込み発生率に比例する数の命令をサンプリングし、その命令についての平均ヒット率を求める点に特徴がある。

【0095】図54(a)及び(b)に、割り込みを考慮した場合の処理の原理を示す。図54(a)において、ヒット率18001は割り込みを考慮しない場合の平均キャッシュヒット率、区間18002、18004はそれぞれヒット率が安定するまでのシミュレーション区間を示し、ヒット率18003、18005はこの区間18002、18004内のヒット率の推移を示す。また、図54(b)は区間シミュレーションが行われた結果、区間19002、19004の平均キャッシュヒット率19003、19005が求まっている状態を示す。

【0096】この区間ヒット率19003、19005がある区間続くものとして処理全体のヒット率、実行時間を見積もることにより、上記第1の実施の形態よりも※50

*レーション処理2006aを行い、各命令でのキャッシュヒット率を含む実行トレース情報D 2007aを出力する。

【0091】出力された実行トレース情報D 2007aと、ストールペナルティ情報 2008とを入力して、各命令毎の実行クロック数を求めて実行トレース情報E 2010aを出力する。この実行トレース情報E 2010aを用いてキャッシュヒット率と実行時間とを見積もる処理2011aを行い、見積もり結果2012aを出力する。

【0092】このような第1の実施の形態によれば、タスクの割り込みを考慮しないキャッシュヒット率と実行時間から、割り込みのプロファイルを基にして、割り込みを考慮したキャッシュヒット率と実行時間が求められる。

【0093】上記第1の実施の形態では、実行される全ての命令について割り込みを考慮したキャッシュヒット率を求める処理を行っている。しかし、各々の命令毎に、ヒット率が安定するに十分な数の命令に対してキャッシュシミュレーション処理を割り込みの種類で決まる回数だけ繰り返すことになるので、非常に長い処理時間が必要となる。以下では、この処理時間の短縮が可能な第2の実施の形態について説明する。

【0094】上記第1の実施の形態では、割り込みが全ての命令において指定された確率で起こると仮定している。しかし、割り込みが発生する確率は通常の場合極めて低い。例えば、クロック周波数50MHzのCPUに対して2m秒毎に割り込みを行う場合を考えると、割り込みの発生確率は、

※高速に値を求めることができる。

【0097】以下に、この原理に基づいて割り込みを考慮した平均キャッシュヒット率、実行時間を見積もる本発明の第2の実施の形態による見積もり方法について説明する。この方法は、区間シミュレーションの処理と実行クロック数を求める処理とを除いて、第1の実施の形態と同様である。

【0098】まず、区間シミュレーションを行う命令のサンプリングを行う。ここでは、全ての割り込みについて予め発生確率が求められていて、割り込みを考慮しない見積もりが既に行われているものと仮定する。

【0099】サンプリングポイントの数は、割り込みの発生率に比例した数が必要である。従って、ある割り込みの発生率を P_i 、全実行命令数を N 、サンプリング点の数を S_i とすると、サンプリング点の数 S_i は、 $N * P_i$ となる。

【0100】サンプリング点は、全実行時間に均等に分布していればよい。そこで、等間隔に抽出するか、あるいは乱数を用いて抽出する。全実行クロック数を NC と

すると、等間隔にサンプリング点を取る場合には、その間隔 K_i は $K_i = NC / S_i$ となる。乱数によりサンプリング点を決定する場合は、例えば0から1の間で一様な乱数を発生する関数RANDを用いて、 $SP = NC * RAND$ という計算を、サンプリングすべき命令の数だけ繰り返す。ここで、SPは実行開始からのクロック数とする。

【0101】サンプリング点が決定した後は、上記第1の実施の形態と同様に区間ヒット率を求める処理を、図40に示されるような手順で行う。処理35002として、割り込みの確率に応じてサンプリング回数を決定し、処理35003として全シミュレーション区間からサンプリング点を決定する。そして、処理35004及び35005として、全てのサンプリング点について区間ヒット率を求める。この区間ヒット率を求める処理は、図27に示す方法で求める。以上の処理35002～35005を、処理35001として示されたように、全ての種類の割り込みに対して行うことで、区間ヒット率が求まる。

【0102】求めた区間ヒット率は、第1の実施の形態と同様にトレース情報D 2007aとして出力される。但し、第1の実施の形態と異なり、図17に示された情報6001～6003を図19における情報8001とすると、図19のようにサンプリング点を示すフラグ情報8002と、区間の長さを示す情報8003とがさらに追加されている。また、区間ヒット率はサンプリング点における命令に対してのみ出力されたものである。

【0103】実行クロック数を求める処理は、図41に示されるようである。処理57001として、処理に必要な最低クロック数NCを設定し、処理57002、57007として分岐によるストールに関する処理を行い、処理57003、57008として命令の組み合わせによるストールに関する処理を行い、さらに処理57006として割り込みの発生に伴うストールに関する処理を行う。これらの処理は、図34に示された処理37001、37002、37007、37003、37008と全く同一である。

【0104】但し、図34に示された処理と異なり、処理を行う対象の命令は処理57010として示されたようにサンプリング点である。処理57011として命令キャッシュヒット率をセーブし、処理57012としてデータキャッシュヒット率をセーブする。さらに、処理57009として示されたサンプリング区間の命令に対しては、処理57013としてセーブされた命令キャッシュ率を用いて実行クロック数NCを求め、処理57014としてセーブされたデータキャッシュ率を用いて実行クロック数NCを求める。この処理によって得られた値を実行トレース情報B2 2007に追加したものが、実行トレース情報C2 2010である。

【0105】ここで、サンプリング区間が重なる場合には、予めキャッシュヒット率を補正しておく必要がある。例えば、図42(a)に示された3つの区間20001～20003は部分的に重複している。そこで、各区間の平均キャッシュヒット率20004～20006に対し、重なり合った区間ではこれらの値の平均をとることで対処する。この結果、図42(b)に示されたように、5つの区間20007～20011に分割され、重なった区間20008～20010における平均キャッシュヒット率は重なった区間のヒット率の平均値となる。実行クロック数を求めた後の処理は、上記第1の実施の形態における場合と同様である。

【0106】上述した第1の実施の形態、又は第2の実施の形態により見積もられるタスクの実行時間及びキャッシュヒット率は、特定のCPUに対してのものである。しかし、例えば図35(a)～(c)にそれぞれ示されたようなキャッシュサイズ/キャッシュ方式が異なる複数のCPUの間でタスクの実行時間及びキャッシュヒット率を求めて相互比較を行ってもよい。

【0107】以下に、このような比較処理を行うものとして、本発明の第3の実施の形態について説明する。本実施の形態により比較処理を行うための手順を、図43及び図44に示す。図43は割り込みを考慮しない見積もりを行ってCPUの比較を行う場合、図44は割り込みを考慮した見積もりを行ってCPUの比較を行う場合を示す。図43において、共通の実行トレース情報A 9001と第1のCPUに関するキャッシュサイズ/キャッシュ方式の情報9002とを用いてキャッシュシミュレーション処理9004を行う。同様に、共通の実行トレース情報A 9001と第2のCPUに関するキャッシュサイズ/キャッシュ方式の情報9003とを用いてキャッシュシミュレーション処理9005を行う。それぞれの処理で得られた情報は、トレース情報B1 9006、トレース情報B2 9007としてそれぞれ出力される。

【0108】以降の処理を、二つのトレース情報B9006、9007に対してそれぞれに行うことで、キャッシュサイズ/キャッシュ方式が異なる第1、第2のCPUの間で、割り込みを考慮しないタスクの実行時間及びキャッシュヒット率の比較を行うことが可能となる。

【0109】同様に、割り込みを考慮したタスクの実行時間及びキャッシュヒット率の比較も行うことができる。割り込みを考慮した場合の処理は、図44に示されるようである。共通の実行トレースA 10001と、第1のCPUに関するキャッシュサイズ/キャッシュ方式に関する情報10002と、割り込み確率データ10003とを用いて区間ヒットシミュレーション処理10006を行い、共通の実行トレースA 10001と、第2のCPUに関するキャッシュサイズ/キャッシュ方式に関する情報10004と、割り込み確率データ10

005とを用いて区間ヒットシミュレーション処理10007を行う。それぞれの処理の結果として、トレース情報B11 1008及びB12 10009とが得られる。ここで、割り込みのプロファイルが同一である場合には、割り込み確率データ10003とデータ1005とは同一となる。

【0110】これ以降の処理は、図43に示された割り込みを考慮しない場合と同様であり、二つのトレース情報B 10008及び10009に対してそれぞれ行うことでキャッシュサイズ/キャッシュ方式が異なる第1、第2のCPU間で割り込みを考慮したタスクの実行時間及びキャッシュヒット率の比較を行うことができる。

【0111】同様の処理を、例えば4種類のキャッシュサイズについて行うことも可能である。図53(a)

(b)(c)にそれぞれ示すように、キャッシュヒット率、実行時間、面積をグラフにプロットする。ここで、第1のキャッシュサイズに関するデータがヒット率58001、実行時間59001、面積60001であり、第2のキャッシュサイズに関するデータがヒット率58002、実行時間59002、面積60002であり、第3のキャッシュサイズに関するデータがヒット率58003、実行時間59003、面積60003であり、第4のキャッシュサイズに関するデータがヒット率58004、実行時間59004、面積60004であるとする。

【0112】システム上で実行される全てのタスクについて、このようなグラフを作成して比較を行うことにより、最適なキャッシュサイズ及びキャッシュ方式のCPUを選択することができる。

【0113】上記第3の実施の形態における処理方法では、キャッシュサイズ及びキャッシュ方式が異なるCPU間での比較を行い最適なCPUの選択を行っている。これに対し、次にCPU以外のハードウェアが変更されたり、見積りの対象となるタスク以外のタスクが変更、又はハードウェア化した場合の見積り方法について、第4の実施の形態として説明する。

【0114】このような変更がおきた場合、対象となるタスクも修正が必要となる可能性があるが、ここでは同一のタスクであると仮定する。この時、対象となるタスクに対して影響があるのは、割り込みのプロファイルが異なってくるという点である。そこで、図2に示された方法で割り込みを考慮しない見積りを行った後、図25に示される割り込みを考慮した見積りを、変更された部分に対応した異なる割り込みプロファイルについて行い、その結果を比較する。

【0115】この方法によれば、見積もる対象のタスク以外の部分が、見積もる対象のタスクに与える影響も見積もることができる。よって、システム上で実行される全てのタスクに対してこのような処理を行うことによ

り、システムの最適化を図ることができる。

【0116】第4の実施の形態による見積り方法は、キャッシュサイズ及びキャッシュ方式が異なる点を除いて他の部分は同一であるCPUに関して比較処理を行うことにより、CPUを選択するものである。これに対し、本発明の第5の実施の形態によれば、CPUの命令セットアーキテクチャが同じで制御方式が異なり、パイプライン段数が異なるCPU間で比較を行うことができる。

【0117】図35(b)及び(c)に、制御方式が異なるCPUの仕様の例を示す。図35(b)に示されたCPU Bは、クロック周波数50MHzで5段パイプライン制御、図35(c)のCPU Cはクロック周波数25MHzで3段パイプライン制御である。この二つのCPU B、CPU Cの各パイプステージにおける動作を図45及び図46に示す。

【0118】CPU Bのパイプライン制御は図45(a)に示されるようであり、命令フェッチIF 47001、命令デコードID 47002、命令実行と実効アドレスの計算EX 47005、メモリへの書き込みMEM 47004、レジスタへの書き戻しWB 47005をパイプラインが備えている。各ステージ47006は、図示されたように重なって実行される。このCPU Bにおけるストールペナルティは、図36(a)に示されるようである。

【0119】図46に、CPU Cのパイプライン制御を示す。命令フェッチ及び命令デコードIF+ID 48001、命令実行及び実効アドレスの計算EX 48002、メモリへの書き込みとレジスタへの書き戻しMEM+WB 48003をパイプラインが含んでいる。各ステージ48004は、重なって実行することができる。このCPU Cにおけるストールペナルティは、図36(b)に示されるようである。

【0120】この二つのCPUを比較すると、図35(b)及び(c)に示されたように、パイプラインの段数とメモリのアクセス時間とは異なるが、命令セット、キャッシュサイズ/キャッシュ方式は同じである。このため、パイプラインストールがおきた場合のペナルティは異なるものであるとしてタスク実行時間を見積もることにより、2つのCPUでの差を見積もることができる。従って、割り込みを考慮しない場合と割り込みを考慮する場合との両方において、異なるストールペナルティを用いて実行クロック数を求めれば良い。

【0121】この実行クロック数を求める処理を図52に示す。トレース情報B 11001は、割り込みを考慮しないシミュレーション処理を行う場合には図2に示されたキャッシュシミュレーション処理1005からの出力1006であり、割り込みを考慮するシミュレーション処理を行う場合は図25に示された区間ヒット率シミュレーション2006の出力2007である。実行ク

ロックを求める処理11004及び11005は、割り込みを考慮しない場合は図2における処理1009の処理と同一であり、割り込みを考慮する場合は図25に示された処理2009と同一となる。

【0122】ストールペナルティ11002として、例えば図36(a)に示された値、ストールペナルティ11003として図36(b)に示された値をそれぞれ設定し、実行クロック数を求めることにより、実行トレース情報C1 11006としてCPU Bについての実行クロック数を求め、実行トレース情報C2 11007としてCPU Cについての実行クロック数を求めることができる。以後の処理は、図1又は図24に示された処理と同様に行うことで、キャッシュヒット率及び実行時間の見積もり値を得ることができる。

【0123】以上のように、命令セットアーキテクチャが同じであって制御方式が異なるCPU間でタスク実行時間を比較することができるため、システムの要求性能にあわせた最適なCPUを選択することが可能である。

【0124】第3、第4及び第5の実施の形態における見積もり結果の比較は、命令セットアーキテクチャが同じCPUに対する比較である。しかし、命令セットアーキテクチャが異なるCPUの間で比較を行うこともできる。このような比較を行う本発明の第6の実施の形態について説明する。

【0125】対象とするCPUとして、第5の実施の形態において説明したCPU Cと、図47に示されるような仕様を有するCPU Dとを対象とする。CPU Dは、CPU Cと命令セットアーキテクチャが異なり、命令語長が命令によって異なり、図49に示されるように1語の命令51001、3語の命令51002、4語の命令51003がある。さらに、データバスが2語同時にアクセス可能なバス幅を有するため、命令のアクセスパターンは図50に示されるように14通りのパターン52001～52014が存在する。ここで、図50において1は1語の命令であり、3-1、3-2、3-3は3語の命令の1語目、2語目、3語目であり、4-1、4-2、4-3、4-4は4語の命令の1語目、2語目、3語目、4語目をそれぞれ示す。

【0126】このCPUは、3段のパイプラインにより制御されている。よって、命令の実行に必要なクロック数は、図51に示されるように3乃至5クロックとなる。命令の実行は、命令フェッチと命令デコードIF+ID 53001、命令実行と実効アドレスの計算EX 53002、メモリへの書き込みとレジスタへの書き戻しMEM+WB 53003の3クロックで行うことが可能である。しかし、命令語長は命令により異なるため、キャッシュヒットしたとしても最悪の場合命令フェッチに3クロックは必要である。

【0127】このようなCPUのストールペナルティは、命令の語数により異なることがあり、図48に示す

ように分岐に伴うストールのペナルティが命令の語調により異なり3から5となる。

【0128】このCPUとCPU Cとを比較する場合、命令セットが異なるためプログラムは異なり、割り込みのプロファイルも異なる。そこで、図2及び図25に示される全ての工程を2つのCPUについてそれぞれ行う必要がある。

【0129】上述した実施の形態はいずれも一例であって本発明を限定するものではなく、必要に応じて様々な変形が可能である。

【0130】

【発明の効果】本発明のタスク実行時間の見積もり方法によれば、割り込みによるタスク実行時間の増加を見積もることができ、これを基にしてCPUの選択、キャッシュサイズ/方式の最適化、さらには、例えば要求される処理能力を満たす範囲内で最小コストのCPUを選別するといったハードウェア/ソフトウェアのトレードオフの決定等を行うことが可能である。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態によるタスク実行時間の見積もり方法における処理の流れを示したフローチャート。

【図2】割り込みを考慮しない場合のタスク実行時間の見積もり方法における処理の流れを示したフローチャート。

【図3】割り込みが発生しない場合のキャッシュヒット率の変化を示したグラフ。

【図4】割り込みが発生した場合のキャッシュヒット率の変化を示したグラフ。

【図5】割り込みによりキャッシュが全て書き換わった場合のキャッシュヒット率の変化を示したグラフ。

【図6】割り込みによりキャッシュの一部が書き換わった場合のキャッシュヒット率の変化を示したグラフ。

【図7】割り込み発生率により実効キャッシュヒット率が変化することを示したグラフ。

【図8】割り込みを考慮した場合におけるキャッシュヒット率の見積もりを示したグラフ。

【図9】命令レベルシミュレーションの処理の流れを示したフローチャート。

【図10】分岐命令の処理の流れを示したフローチャート。

【図11】ロード命令の処理の流れを示したフローチャート。

【図12】セーブ命令の処理の流れを示したフローチャート。

【図13】レジスタ間演算命令の処理の流れを示したフローチャート。

【図14】トレース情報Aに含まれる各々命令の内容を示した説明図。

【図15】トレース情報Aに、命令の組み合わせによっ

て発生するストールを示すフラグを加えたことを示す説明図。

【図16】トレース情報Aに、命令とデータのアクセス回数を加えたことを示す説明図。

【図17】トレース情報Bに含まれる各々命令の内容を示した説明図。

【図18】トレース情報Cに含まれる各々命令の内容を示した説明図。

【図19】トレース情報Bにサンプリング点と区間の長さを加えたことを示す説明図。

【図20】キャッシュメモリの構造を示した説明図。

【図21】割り込みを考慮することなくキャッシュシミュレーションを行うためのデータ構造を示した説明図。

【図22】キャッシュシミュレーションの処理の流れを示したフローチャート。

【図23】キャッシュリードアクセスの処理の流れを示したフローチャート。

【図24】キャッシュライトアクセスの処理の流れを示したフローチャート。

【図25】割り込みを考慮した場合のタスク実行時間の見積もり方法における処理の流れを示したフローチャート。

【図26】区間ヒット率を求める処理の流れを示したフローチャート。

【図27】区間ヒット率を求める処理の流れを示したフローチャート。

【図28】割り込みプロファイルを示した説明図。

【図29】割り込みにおけるキャッシュ書き換わりパターンAを示した説明図。

【図30】割り込みにおけるキャッシュ書き換わりパターンBを示した説明図。

【図31】区間ヒット率を求める処理において、キャッシュシミュレーションを行うためのデータ構造を示した説明図。

【図32】区間ヒット率を求める処理において、キャッシュリードアクセスを行う処理の流れを示したフローチャート。

【図33】区間ヒット率を求める処理において、キャッシュライトアクセスを行う処理の流れを示したフローチャート。

【図34】割り込みを考慮したシミュレーションにおいて実行クロック数を求める処理の流れを示したフローチャート。

【図35】CPU A～Cの仕様を示した説明図。

【図36】CPU A～Cのストールペナルティを示した説明図。

【図37】割り込みを考慮しないシミュレーションにおいて実行クロック数を求める処理の流れを示したフローチャート。

【図38】割り込みを考慮しないシミュレーションにお

いてキャッシュヒット率、実行時間見積もりを行う処理の流れを示したフローチャート。

【図39】割り込みを考慮したシミュレーションにおいてキャッシュヒット率、実行時間見積もりを行う処理の流れを示したフローチャート。

【図40】本発明の第2の実施の形態によるタスク実行時間の見積もり方法における区間ヒット率を求める処理の流れを示したフローチャート。

【図41】同第2の実施の形態によるタスク実行時間の見積もり方法において実行クロック数を求める処理の流れを示したフローチャート。

【図42】本発明の第3の実施の形態によるタスク実行時間の見積もり方法において割り込みによる影響を考慮すべき区間が重なっている場合の処理を示した説明図。

【図43】同第3の実施の形態によるタスク実行時間の見積もり方法において、キャッシュサイズとキャッシュ方式が異なるCPUの比較を行う処理を示したフローチャート。

【図44】同第3の実施の形態によるタスク実行時間の見積もり方法において、キャッシュサイズとキャッシュ方式が異なるCPUの比較を行う処理を示したフローチャート。

【図45】CPU A及びCPU Bのストールペナルティを示した説明図。

【図46】CPU Cのパイプラインステージを示した説明図。

【図47】CPU Dの仕様を示した説明図。

【図48】CPU Dのストールペナルティを示した説明図。

【図49】CPU Dの命令の構成を示した説明図。

【図50】CPU Dの命令アクセスパターンを示した説明図。

【図51】CPU Dのパイプラインステージを示した説明図。

【図52】本発明の第5の実施の形態によるタスク実行時間の見積もり方法により、ストールペナルティが異なるCPUについて比較を行う処理の流れを示したフローチャート。

【図53】キャッシュサイズの相違により、キャッシュヒット率、実行時間、面積が異なることを示したグラフ。

【図54】本発明の第2の実施の形態におけるキャッシュヒット率の見積もりを示した説明図。

【符号の説明】

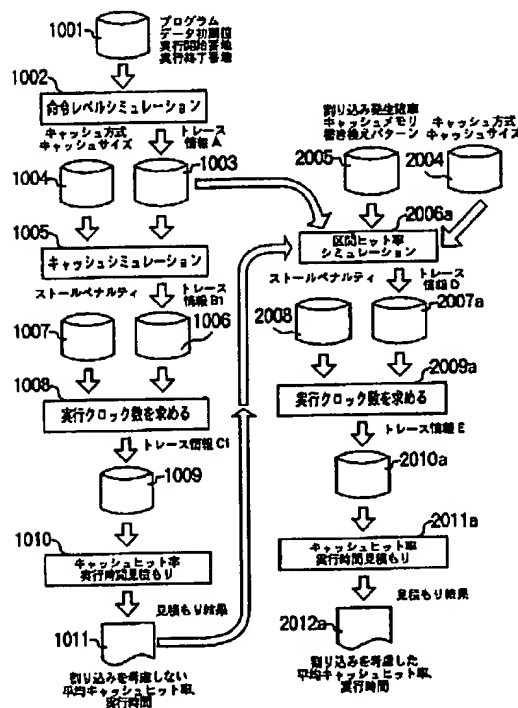
12001、12002、13001～13004、14001～14005、15001～15005、16001～16004、18001～18005、19001～19005 キャッシュヒット率

1001、2001 プログラム、データ初期値、実行開始番地、実行終了番地

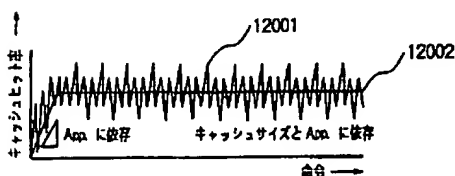
29

1002、2002 命令レベルシミュレーション
 1003、2003、9001、10001 トレース
 情報A
 1004、2004、9002、9003、1000
 2、10004 キャッシュ方式、キャッシュサイズ
 1005、9004、9005 キャッシュシミュレー
 ション
 2005、10003、10005 割り込み確率デー
 タ
 2006、2006a、10006、10007 区間 10
 ヒット率シミュレーション
 1006、9006、10008 トレース情報B1
 2007、2007a、9007、10009 トレー
 ス情報B2
 11001 トレース情報
 1007、2008、11002、11003 ストー
 ルペナルティ
 1008、2009、2009a、11004、110

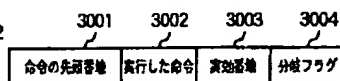
【図1】



【図3】



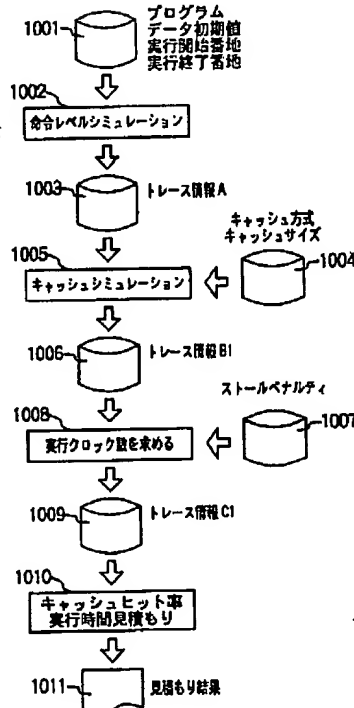
【図14】



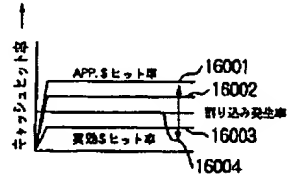
30

05 実行クロック数演算処理
 1009 トレース情報C1
 2010、2010a トレース情報C2
 1010、2011、2011a キャッシュヒット率
 実行時間見積もり処理
 1011、2012、2012a 見積もり結果
 3001 命令先頭番地
 3002 実効命令
 3003 実効番地
 3004 分岐フラグ
 4002 ストールフラグ
 5002 命令アクセス回数
 5003 データアクセス回数
 6002 命令キャッシュヒット回数
 6003 データキャッシュヒット回数
 7002 命令実行に必要なクロック数
 8002 サンプリング点フラグ
 8003 区間の長さ

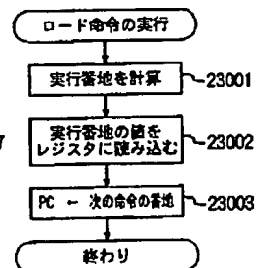
【図2】



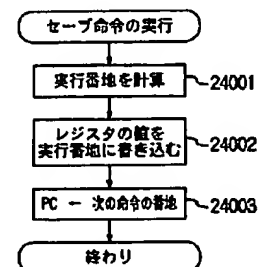
【図7】



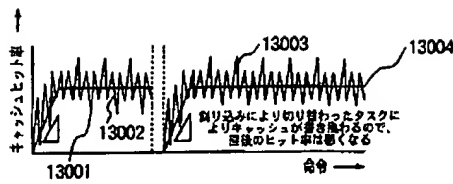
【図11】



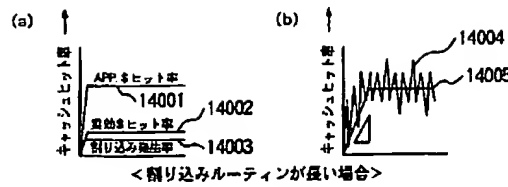
【図12】



【図4】



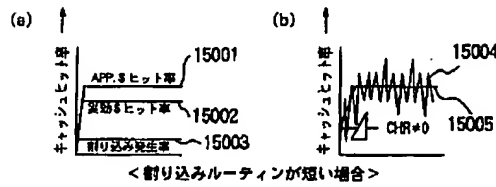
【図5】



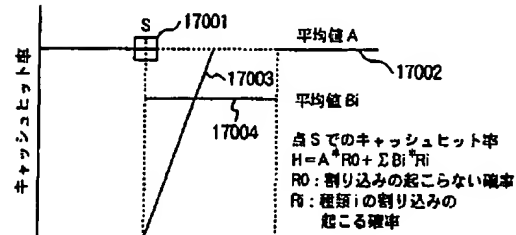
【図48】

NBR	3~5
NISTL	0
NIF	4
NDATA	4

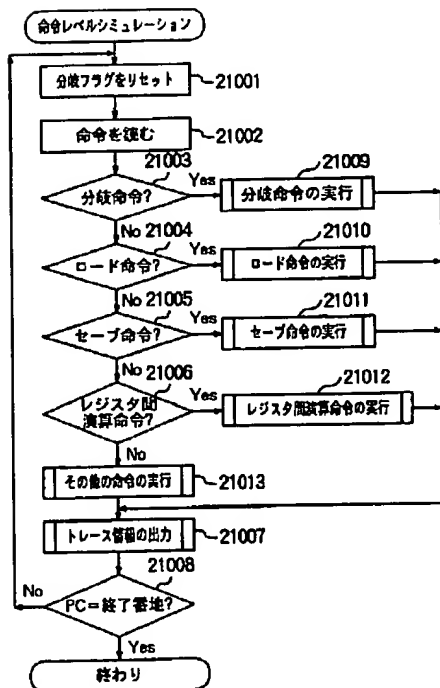
【図6】



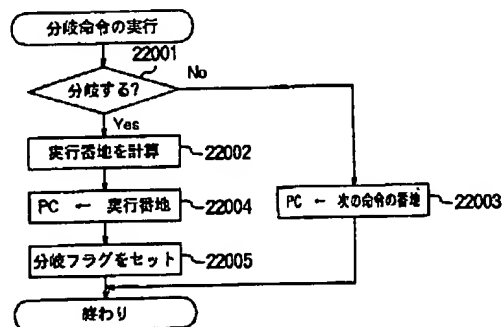
【図8】



【図9】

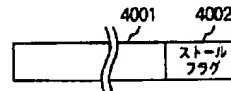
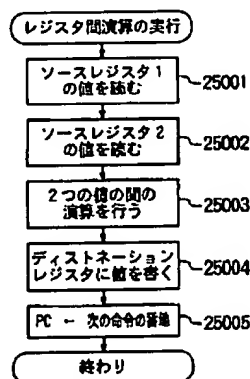


【図10】

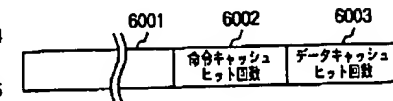


【図13】

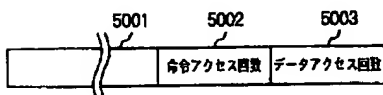
【図15】



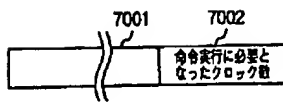
【図17】



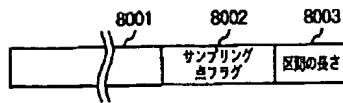
【図16】



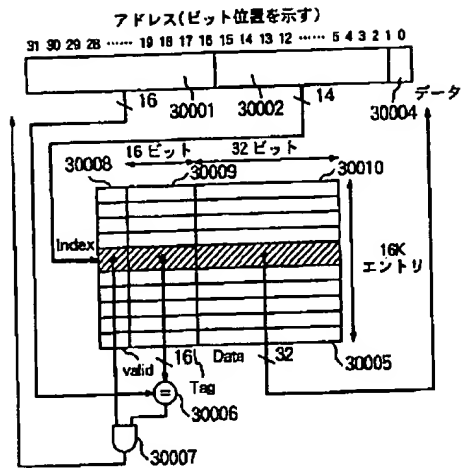
【図18】



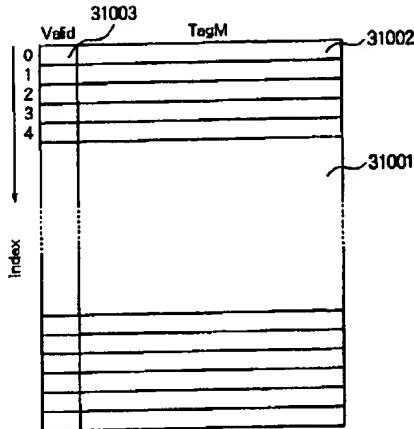
【図19】



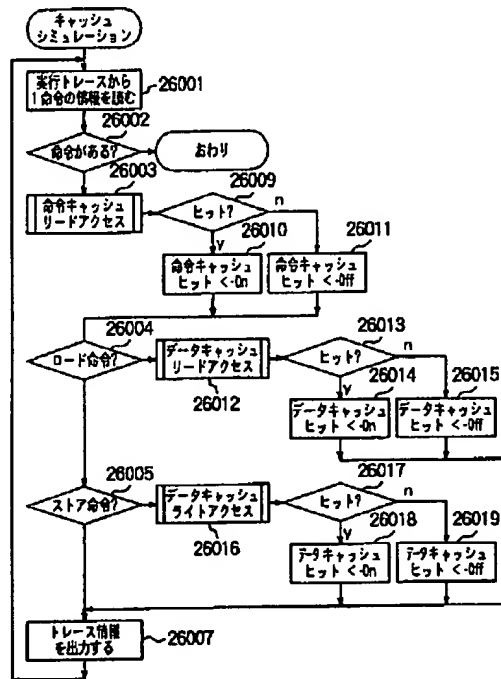
【図20】



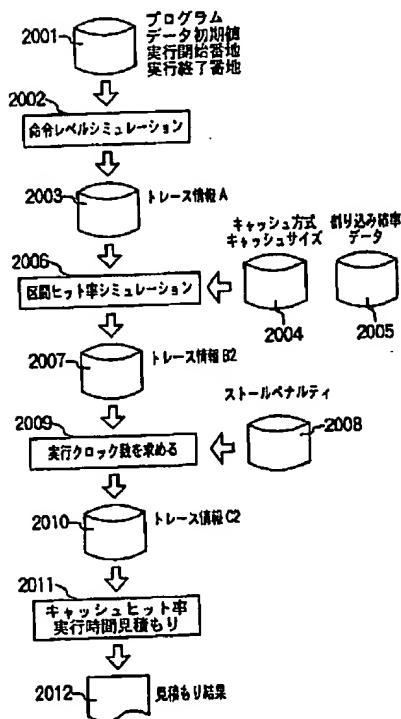
【図21】



【図22】



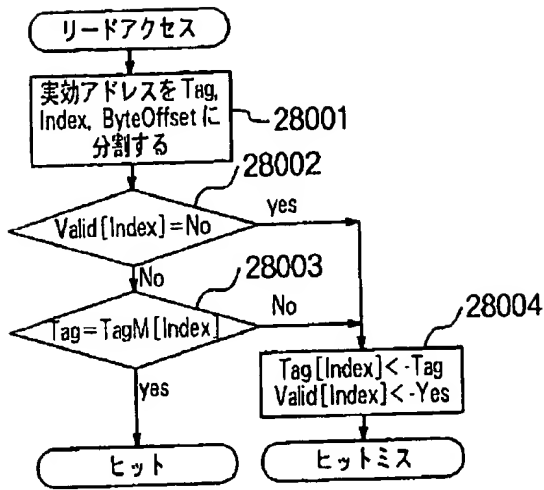
【図25】



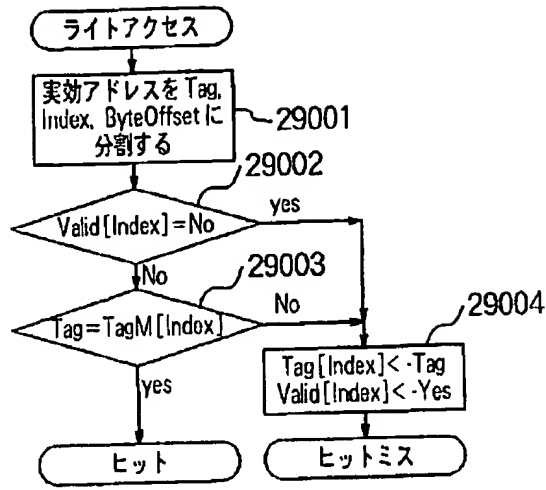
【図28】

割り込み種別	発生確率	キャッシュパターン
A	2.0×10^{-5}	30%クリア
B	1.5×10^{-6}	パターン A
C	2.0×10^{-7}	パターン B/D/110%
D	2.0×10^{-8}	100%クリア
E	1.0×10^{-7}	80%クリア

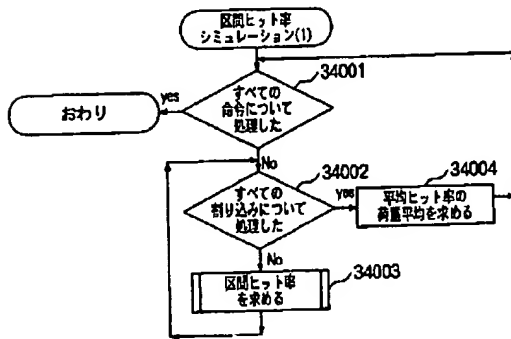
【図23】



【図24】



【図26】



【図29】

パターン A	
命令キャッシュ	データキャッシュ
000	000
5	5
03F	00F
100	030
5	5
11F	03F
—	100
—	5
—	110

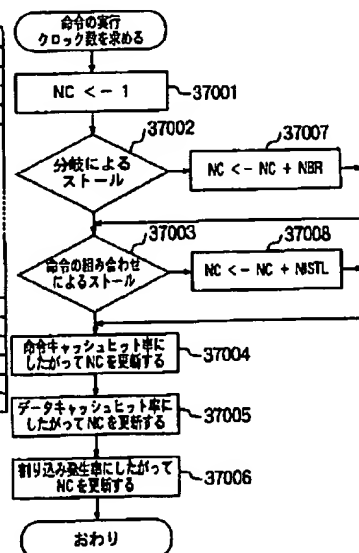
【図30】

パターン B	
命令キャッシュ	データキャッシュ
000	000
5	5
05F	03F
080	100
5	5
0FF	1FF
500	—
5	—
5FF	—

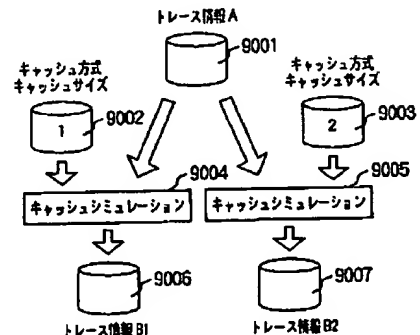
【図31】

	Valid	Valid0	TagM
0			
1			
2			
3			
4			
Index ↓			
54001			
54002			
54003			

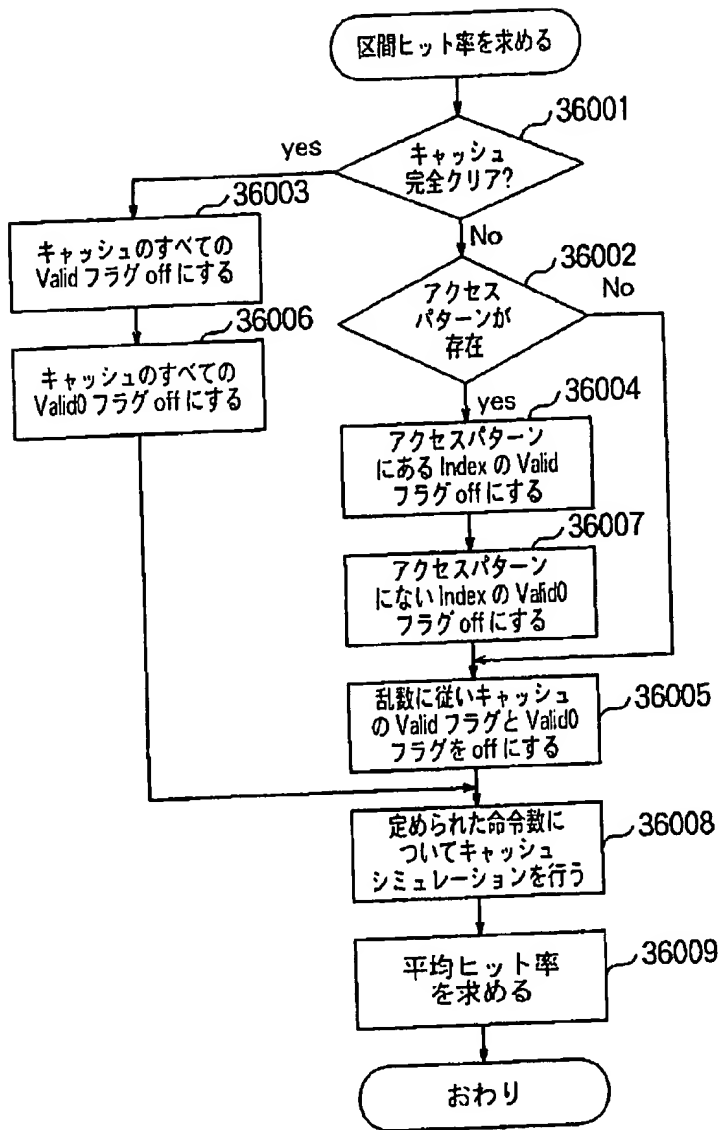
【図34】



【図43】



【図27】



【図36】

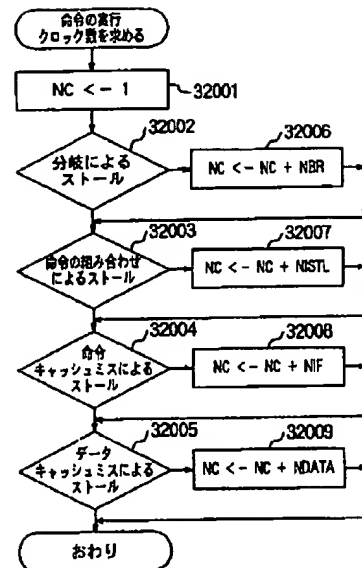
(a)

NBR	5
NISTL	0
NIF	3
NDATA	3

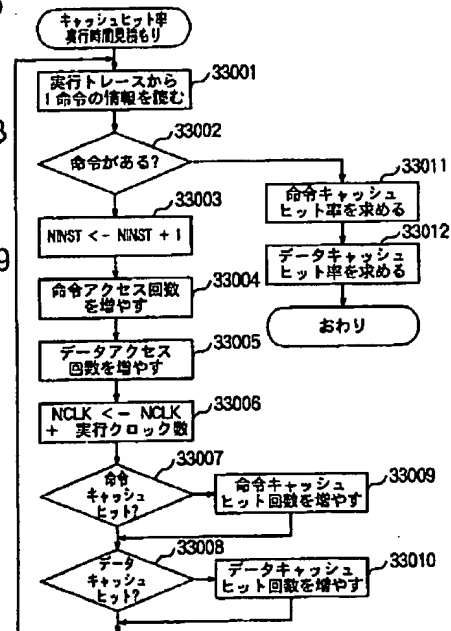
(b)

NBR	3
NISTL	0
NIF	4
NDATA	4

【図37】



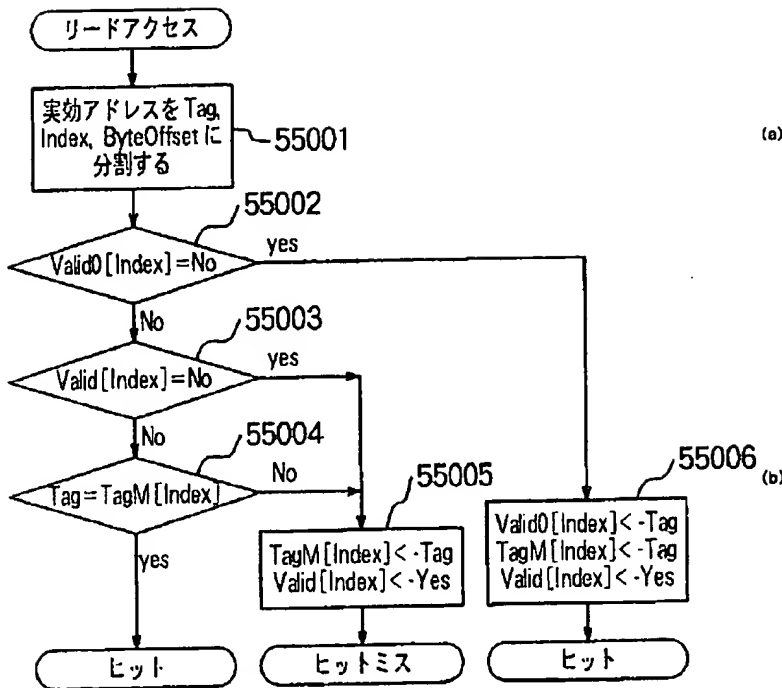
【図38】



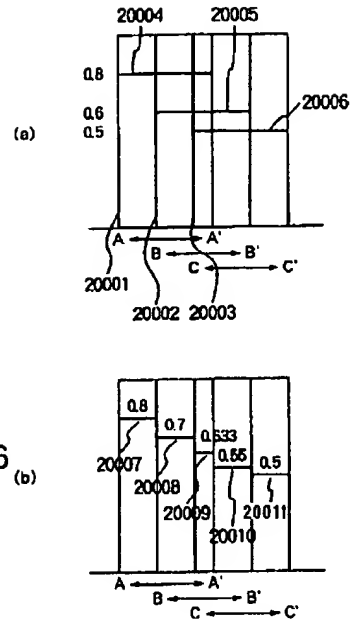
【図47】

CPU D	3
パイプライン段数	25MHz
クロック周波数	成立 3~5 / 不成立 0
分岐ペナルティ	100nsec
命令キャッシュミス・ペナルティ	100nsec
データ	ダイレクトマップ 512 entry
命令キャッシュ	ダイレクトマップ 512 entry
データキャッシュ	

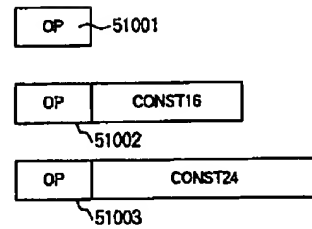
【図32】



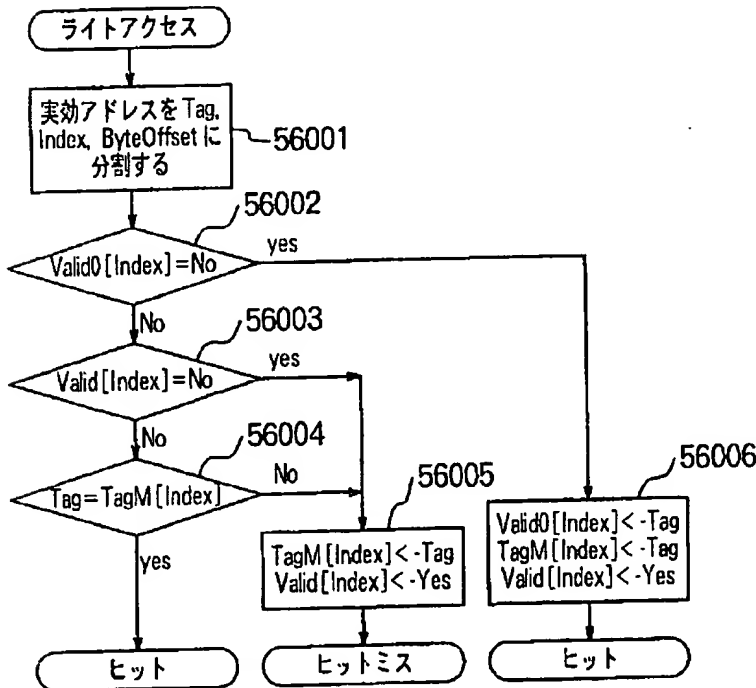
【図42】



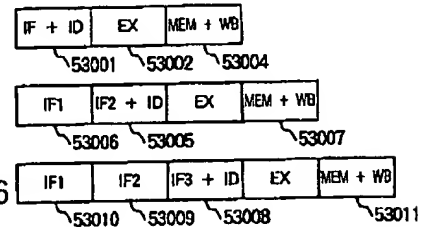
【図50】



【図33】



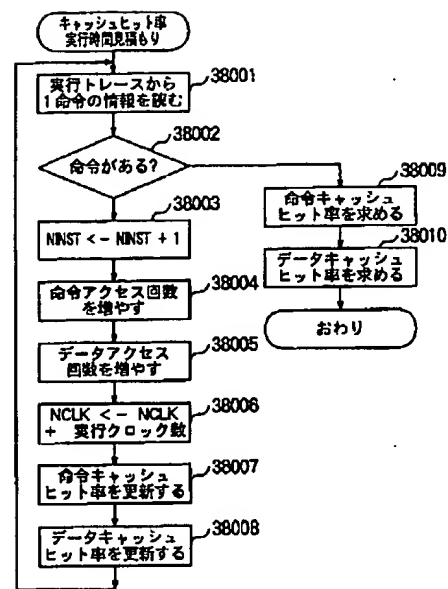
【図51】



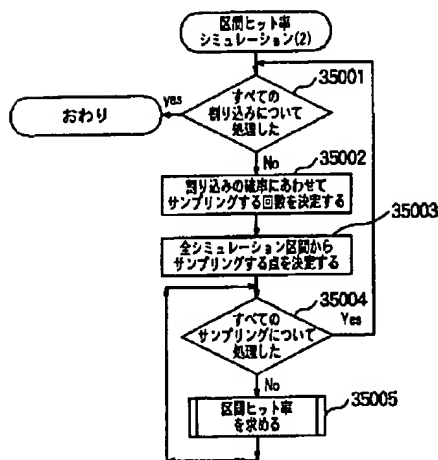
【図35】

CPU A	
パイプライン段数	5
クロック周波数	50MHz
分岐ペナルティ	成立 5 / 不成立 0
命令キャッシュミス・ペナルティ	60nsec
データ	60nsec
命令キャッシュ	ダイレクトマップ 4096
データキャッシュ	ダイレクトマップ 4096
CPU B	
パイプライン段数	5
クロック周波数	50MHz
分岐ペナルティ	成立 5 / 不成立 0
命令キャッシュミス・ペナルティ	60nsec
データ	60nsec
命令キャッシュ	ダイレクトマップ 1024
データキャッシュ	ダイレクトマップ 1024
CPU C	
パイプライン段数	3
クロック周波数	25MHz
分岐ペナルティ	成立 3 / 不成立 0
命令キャッシュミス・ペナルティ	100nsec
データ	100nsec
命令キャッシュ	ダイレクトマップ 1024
データキャッシュ	ダイレクトマップ 1024

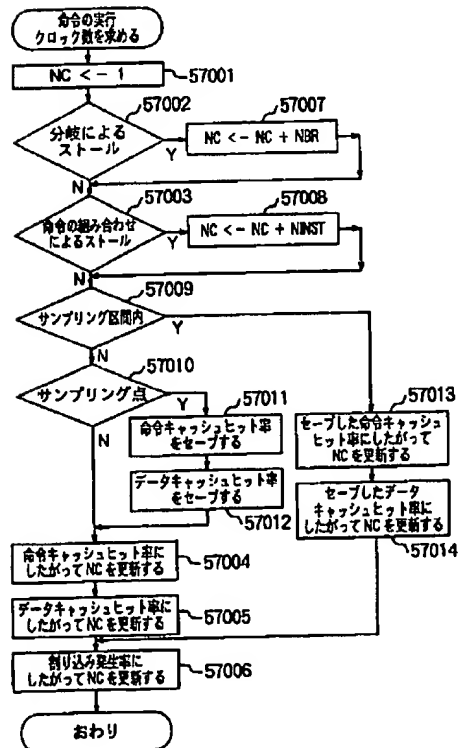
【図39】



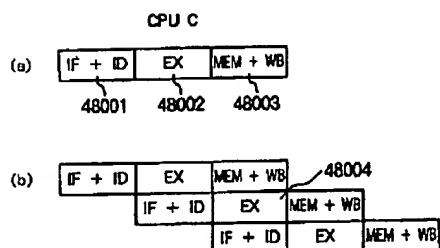
【図40】



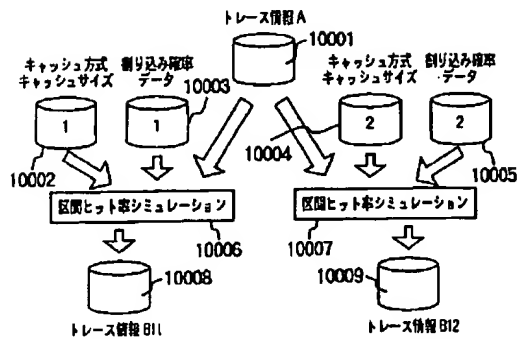
【図41】



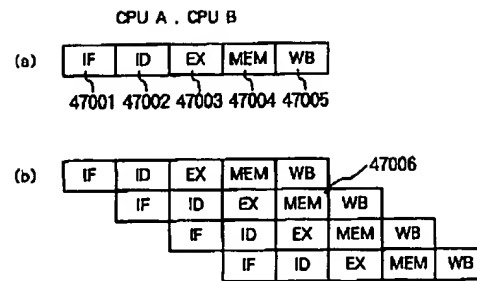
【図46】



【図44】



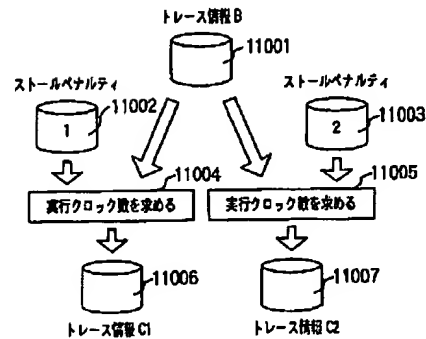
【図45】



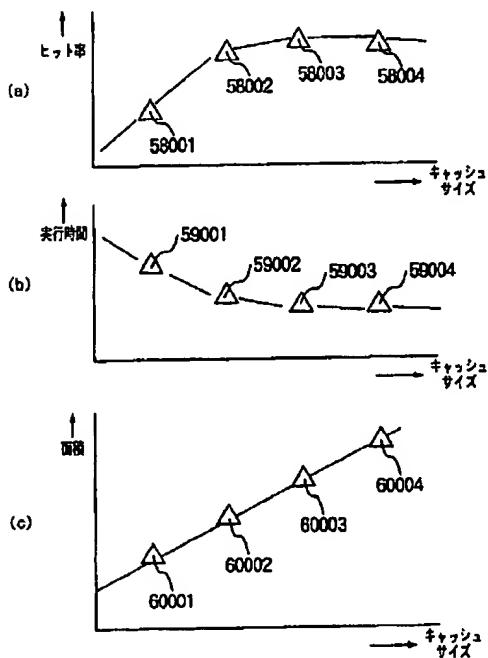
【図49】

1	1	52001	3-3	4-1	52008
1	3-1	52002	4-1	4-2	52009
1	4-1	52003	4-2	4-3	52010
3-1	3-2	52004	4-3	4-4	52011
3-2	3-3	52005	4-4	1	52012
3-3	1	52006	4-4	3-1	52013
3-3	3-1	52007	4-4	4-1	52014

【図52】



【図53】



【図54】

